

II. BRIEF DESCRIPTION OF THE PRESENT INVENTION

A. Overview

Applicant has invented a software development system for dynamic web applications running as a *web application* inside a web browser. The inventive system can be used without client based software installation. The server based architecture allows the developer to see and work on pages generated by the web application under development. These closely match the pages seen by an end-user of the application. In contrast, conventional development systems show a page template during development. Because page templates do not show dynamically generated content, a page template can look very different than the page generated from it, i.e., the page shown to the end-user. The present invention ensures that, although the user edits the generated page, the page template on the server is updated accordingly.

The editor denotes a component on the pages using an XML-element. HTML code generation works dynamically when a page is requested. Advantageously, since XML elements can be nested, so can the interactive server side components disclosed in the present invention.

B. Editor

Conventional server based dynamic web pages usually show different content and layout with every page display. Technically, this is realized using page templates or page generating programs (called component pages in applicant's disclosure – see Fig. 7) that run on the server and produce generated pages for every display request by the browser.

Many prior art approaches, including Massena, provide client-side based editors for such page templates. In addition, Massena provides a way to display the generated page in a web browser. The present invention, on the other hand, allows the user to edit the generated page itself directly in the browser. Changes made by the user are then sent back to the server and applied to the page template by the server part of the editor. (Compare Fig. 1 of Massena to Fig. 18 of the present disclosure).

The editor of the present invention displays a single window on screen that shows the page in the form of the generated page, but not a page template, as those terms are used above and in the present disclosure. (note: the editor uses multiple windows but just one for displaying

a view of the page). In accord with the present invention, the user can edit the generated page directly by clicking on the handle of a component. According to Fig. 1 of Massena, the client computer has an editor and a browser, with the editor displaying the page template and the browser displaying the generated page. In order to modify the generated page (which is displayed in the browser), the user needs to switch back to the editor, find the correct position in the page template that corresponds to the place in the generated page he wants to modify, and then do the modification. This process becomes more tedious as the differences increase between the generated page (shown in the browser) and the page template (shown in the editor) and as applications become more complex.

In accord with the WYSIWYG principle as applied to *applications*, it would be desirable if the page displayed during editing *worked as similarly* as possible to the end user's application. The present invention achieves this by editing the generated page. If Massena is used for dynamic pages, it seems to be the responsibility of each component to display a design-time view that is similar to the run-time view. This appears to applicant to be difficult (or even impossible) for server side components, especially if the components behaviour depends on data available on the server only. Interactive Server Side Components ("ISSC") furthermore can dynamically hide parts of the page template. So for example multilingual pages can be built by hiding all unwanted language texts. In such an example the page template differs substantially from the generated page: the page template contains intermingled text of all languages while the generated page correctly displays just a single selected language.

The inventive editor is a server based web application. It is installed on a server and can be used by every client computer connected to this server (assuming the client computer has a javascript enabled browser installed - see Fig. 18). Explicit installation on the client computers is not required. This means that the editor can be used for server based business models as well as in groupware and content management applications. Editing directly in the browser also allows the user to make sure that the pages being edited stay fully functional during editing. In contrast, Massena requires an editor program running in a client based environment and so it must be installed on every client computer.

It is also useful to run the inventive editor together with the browser and a web server on a single computer to just take advantage of editing generated pages.

The central idea of the inventive editor is that during editing, the same generation steps are performed as during normal execution of the web application, and in addition, editing features are inserted into the page. This ensures that the developer works on a page (the generated page) that functions just like the page seen by a normal end user of the web site (with the exception of the editing features).

Editing features include a user interface for editing, information regarding the components on the page template for display by the editor, and information regarding merging the changes made by the user back into the page template. Whenever the user modifies a component, the editor sends the change to the server and modifies the page template accordingly (based on the information collected during page generation).

C. Components

When building applications from components, flexibility is very important. With enhanced flexibility, more components can be combined, and more diverse applications can be built without resorting to scripting. Therefore, applications can be better adapted to users' needs. The components of the present invention are dynamic while the components disclosed by Massena are static. By dynamic, we mean that the components displayed on the generated page can differ with every page request in number, size or other characteristic, from the components initially included on the page template.

Massena generates code inside the editor at design time and inserts it into the page template. Therefore, design time controls ("DTCs") are a part of Massena's editor (see Massena Fig. 1) and run on the client side during editing. In contrast, the inventive editor inserts tags into the page template marking the positions of dynamic components. Components are generated by the page generator running at request time. Therefore, interactive server side components run at request time as part of the page generator on the server (see present disclosure, Fig. 7). A component is generated only when the page generator actually executes it, i.e., at page generation time it is decided to generate zero, one, or multiple component instances and generation can take advantage of the data available at request time. Components can be interactive and wait for user interaction in subsequent page requests, retaining access to their data.

For example, a list of database records might contain a delete-link component to delete the database record. The delete-link component exists multiple times on the generated page (once for each database record displayed). However, each instance of the link component refers to a different database record. The inventive component mechanism makes sure that the process method of the correct delete-link component is called if the link is clicked. The data of that component is passed in the session memory of the server thus avoiding tampering with the data on the client side. Additionally, these interactive server side components make sure that the process method is not called more than once due to reloading the page.

Components are built to cooperate with the editor by providing information for editing and by displaying handles when used in edit mode. There is no separate design-time and run-time view of a component as in Massena; instead a component is, except for the additional handles, displayed during editing the same way as at run time. This reduces the programming overhead of components, since just one view needs to be implemented.

III. BRIEF DESCRIPTION OF THE CITED REFERENCE

DTCs (Design Time Controls) are an extension of a client side editor (Massena figure 1). A client side editor clearly distinguishes between design-time and run-time. During design-time, the developer uses the editor program to create a web page template in form of a text file. This text file is uploaded to a server. Afterwards, at run-time, the page can be requested and displayed by a web browser. DTCs run at design-time of a web page as part of the editor and generate text into the HTML page template created (see Massena abstract). Because of the generation at design time, applicant submits that the number and characteristic of all components generated into the page is static and cannot change on a per request basis. As long as DTCs generate static HTML, pages created with Massena's approach appear to be static pages, not dynamic web applications.

DTCs also can generate dynamic content (e.g. server side scripts) into pages, which makes them page templates. At run time, dynamic content is evaluated, i.e. the page template is transformed into a generated page (either on the server or inside the browser).

Client side editors work nicely for static pages but have problems with pages generated dynamically. Dynamic pages may change with every request, while the editor shows a static

document. Dynamic pages change based on the execution of a program on the server. Because the client based editor does not execute the program, it therefore cannot display the page as the end user would see it.

IV. CLAIMS 1-40 ARE PATENTABLE OVER THE CITED REFERENCE

The Examiner rejected claims 1-40 under 35 USC §102(a) as anticipated by U.S. Patent No. 6,035,119 (“Massena”). However, applicant respectfully traverses the rejection for the reasons discussed below.

Claim 1

Claim 1 is an independent claim directed to a software development system having a page generator and an editor. The Examiner cites Fig. 1 and the Abstract of Massena in support of his assertion that Massena identically discloses the invention recited in applicant’s claim 1, namely

a page generator capable of generating functional application pages with additional editing features for interpretation by the browser program

an editor capable of directly operating on the pages displayed by the browser

(Paper No. 4 at 2). However, applicant submits that the Examiner is in error in reaching that conclusion. The cited portions of Massena do not teach or suggest “*editing features for interpretation by the browser program*” as claimed. Instead, Massena discloses the ability to edit pages only at “design-time” using an editor separate from the browser. In fact, Massena clearly states that generated pages may contain both design-time information and run-time information, but interpreted by a browser program, that design-time information is hidden and only run-time information is interpreted. For example, in column 5, lines 4-13, Massena states that design-time information is ignored by the browser and, in fact, the browser ignores everything but run-time information. In contrast, applicant’s claim 1 specifically recites that the page generator generates functional application pages with additional editing features for interpretation by the browser program. Therefore, applicant submits that Massena does not teach or suggest *additional editing features for interpretation by the browser program*, as recited in claim 1.

The examiner further asserts that Massena (see Fig. 1, Abstract) discloses an editor capable of directly operating on the pages displayed by the browser. However, applicant submits

that the examiner is in error reaching that conclusion. In conventional development systems, like Massena, the file displayed in the editor and the generated page displayed in the browser can differ. For example, Massena states that the file edited by the editor is transformed before being saved to the server. (Massena at column 4, lines 66-67, and Fig. 1). Also, Massena states that in the case of server based scripting, the page generator on the server transforms the page before sending it to the browser. (Massena at column 5, lines 8-9). In contrast, claim 1 specifically recites the fact that the editor directly operates on the pages displayed by the browser. Therefore, applicant submits that Massena does not teach or suggest this feature.

Claim 2

Claim 2 is dependent from claim 1. For all the reasons discussed regarding claim 1 above, claim 2 is likewise not taught or suggested by Massena.

Claim 3

Claim 3 is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claim 1 above, claim 3 is likewise not taught or suggested by Massena.

In addition, the examiner asserts that Massena at column 3, lines 30-37 and column 5, lines 1-11, discloses *at least one of the components reacts interactively on user input by executing instructions on the server*, as recited in claim 3. However, applicant's reading of the cited portions reveals no support for the examiner's position. Instead, the cited portion in column 3 appears to disclose the manner in which DTCs (Design Time Controls) are integrated into the editor running on the client computer at design-time. In contrast, applicant's claim 3 is specifically concerned with actions on the server at run-time. Likewise, the cited portion in column 5 of Massena appears to disclose how to save and transform a page by inserting server side scripting. Massena does not elaborate on the scripts generated. However, in conventional development systems, server side scripts are normally executed on page generation taking place **before** display of a component. In contrast, claim 3 recites components that, upon user interaction (which necessarily happens **after** display of the component), react by executing instructions on the server. This feature is neither taught nor suggested by Massena.

Claim 4

Claim 4 is dependent from claim 3, which is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claims 1 and 3 above, claim 4 is likewise not taught or suggested by Massena.

In addition, the examiner asserts that Massena discloses one component contained within another component and refers to the “component objects” discussed at column 5, lines 25-35 of Massena. (Paper No. 4 at 3). However, applicant’s reading of the cited portion reveals no support for the examiner’s position. Instead, the cited portion appears to disclose that DTCs can be used on multiple pages, thus replicating text, and that DTCs can be used to generate various versions of the text for different browsers. Further, the cited portion seems to disclose that repeatable web page content can be encapsulated inside component objects. In contrast, applicant claims that components can contain other components, i.e., according to the definition of “contains” at page 10, lines 5-8 of the application as filed, other components can be written between the begin and end tag of a component, which is very different from a component being contained on multiple pages or that repeatable web page content can be encapsulated inside a DTC. For these reasons, applicant submits that Massena does not teach or suggest a software development system *wherein at least one of the components contains at least one other component*, as recited in claim 4.

Claim 5

Claim 5 is dependent from claim 3, which is dependent from claim 2, which is dependent from claim 1. For all the reasons discussed regarding claims 1 and 3 above, claim 4 is likewise not taught or suggested by Massena.

The examiner says that Massena 5:35-37 discloses the fact that the set of components displayed on a pages generated from a single page template can vary for different requests of the same page. However, applicant’s reading of Massena at column 5:35-37 reveals no support for the examiner’s position. Instead, the cited text appears to disclose that DTCs can be used to generate various versions of the text for different browsers. In contrast, applicant claims that the set of components can change, i.e., additional components can appear or existing components can vanish, which is quite different from a DTC being displayed differently in different browsers.

Applicant has amended claim 5 to remove the word “display” from the claim as unnecessary since the pages generated have components, regardless of whether those components are actually displayed in the browser.

Claim 6

Claim 6 is an independent claim directed to a software development system having a page generator and an editor. Applicant has amended claim 6 as shown in Appendix A, not for the purpose of distinguishing the cited reference, but for increased clarity with regard to the interactive nature of the components. Claim 6 imposes specific limitations on the server computer, as follows:

a data store,

a plurality of components residing in the data store, including components that react interactively on user input by executing instructions;

a plurality of page templates residing in the data store, at least one page template having at least one selected component incorporated therein; and

a server processor controlled by a third software program, said program providing instructions for selecting a page template based on the request from the client computer and instructions for generating a page from the page template for transmission to the client computer.

The examiner asserts that Massena discloses a data store on the server and cites column 8, line 7 and column 9, lines 55-60 in support of his assertion. However, applicant submits that the examiner is in error reaching that conclusion. In fact, the registration database and the persistence interface described in Massena are located on the client computer running the editor, not on the server computer. This is clear from Fig. 1 and the abstract of Massena, which indicate that the design time control, and consequently the persistence interface and the registration database used to identify a DTC, are running at design-time as part of the editor on the client.

The examiner also asserts that Massena discloses a plurality of components residing in the data store on the server, citing column 8, lines 1-10 in support of his assertion. However, as noted above, the registration database and the persistence interface of Massena are located on the

client computer running the editor, not on the server computer. Further, there is nothing in the cited portion that teaches or suggests that the components react interactively.

Further, the examiner cites column 3, lines 15-20 and Fig. 1 of Massena as disclosing a server processor controlled by a third software program on the server. The cited portion of Massena discloses other ways to create DTCs, but does not talk about programs running on the server. Fig. 1 of Massena appears to indicate that page templates are stored on the web server, including run time text but not a server processor.

Claim 7

Claim 7 is dependent from claim 6. For all the reasons discussed regarding claim 6 above, applicant submits that claim 7 is not taught or suggested by Massena.

Claim 8

Claim 8 is dependent from claim 6. For all the reasons discussed regarding claim 6 above, applicant submits that claim 8 is likewise not taught or suggested by Massena.

In addition, the examiner asserts that Massena discloses one component contained within another component. However, as previously noted with regard to claim 4 above, Massena fails to teach or suggest such a feature.

Claim 9

Claim 9 is an independent claim directed to a method for generating documents that react interactively. The examiner asserts that Massena discloses the assignment of a unique identifier for a component and embedding the identifier into a generated page, citing column 7, line 60 to column 8, line 10 in support of his assertion. However, applicant submits that the examiner is in error in reaching that conclusion. The component identifier discussed in Massena is actually an identifier for component classes, not the component itself. For example, two counters on a single page would receive a single identifier according to Massena, while in the present invention they would receive two identifiers. This becomes clear by reviewing Massena at column 8, line 2 et seq., which explicitly talks about class identification. In contrast, applicant's claim expressly

recites that unique identifiers are assigned to the component itself, not to a class of components as in Massena.

Claim 9 was amended to emphasize that the unique identifiers are generated dynamically upon request for a page by the browser and for enhanced clarity with regard to the interactive nature of a component. In contrast, Massena generates pages statically.

Claim 10

Claim 10 is dependent from claim 9. For all the reasons discussed with reference to claim 9, claim 10 is also patentably distinct from the cited reference.

The examiner states that column 11, lines 58-60 of Massena discloses hosting web pages for clients and asserts that storing data is inherent to hosting web pages. While applicant does not refute that statement, it misses the point. Applicant's claim recites that data is stored on a server for at least one of the components, which indicates that the data store is organized in a way that facilitates the storing and retrieving of information on a per component basis. This is simply not the case for web hosting that just stores pages.

The claim has been amended replacing "for" by "representing" to enhance clarity.

Claim 11

Claim 11 is dependent from claim 10, which is dependent from claim 9. For all the reasons discussed with reference to claims 9 and 10, applicant submits that claim 11 is patentably distinct from the cited reference. Claim 11 has been amended consistent with the amendment of claim 9.

In addition, the examiner asserts that Massena discloses a method of analyzing a request sent by the browser for unique identifiers, citing column 7, lines 60-67 and column 8, lines 1-10 in support of his assertion. However, applicants reading of the cited portion of Massena reveals no support of the examiner's position. Instead, the cited portion appears to describe identifying DTCs during editing on the client computer, which appears irrelevant to analyzing a request sent by the web browser. For this additional reason, applicant submits that claim 11 is patentably distinct from the cited reference.

Claim 12

Claim 12 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 12 is patentably distinct from the cited reference.

Claim 13

Claim 13 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 13 is patentably distinct from the cited reference.

Claim 14

Claim 14 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 13 is patentably distinct from the cited reference.

In addition, the examiner asserts that Massena discloses a method where one component calls another component and refers to the “component objects” discussed at column 5, lines 25-35 of Massena. (Paper No. 4 at 3). However, as previously discussed, applicant’s reading of the cited portion reveals no support for the examiner’s position. Instead, the cited portion appears to disclose that DTCs can be used on multiple pages, and that DTCs can be used to generate various versions of the text for different browsers. Further, the cited portion seems to disclose that repeatable web page content can be encapsulated inside component objects. Applicants understanding of encapsulation is that DTCs can generate the same web page content several times, thus replicating the text. In contrast, applicant’s claim asserts that at least one component calls another component, which is very different from a component being contained on multiple pages and also different from the fact that repeatable web page content can be encapsulated inside a DTC. For these reasons, applicant submits that Massena does not teach or suggest a software development system *wherein at least one of the components calls at least one other component*, as recited in claim 14.

Claim 15

Claim 15 is dependent from claim 11, which is dependent from claim 10, which is in turn dependent from claim 9. For all the reasons discussed with reference to claims 9-11, applicant submits that claim 13 is patentably distinct from the cited reference.

Claim 16

Claim 16 is an independent claim directed to a method for implementing client server applications. The examiner asserts that Massena discloses a method that involves storing data objects on a server, and cites column 8, line 7 and column 9, lines 55-60 in support of his assertion. However, applicant submits that the examiner is in error reaching that conclusion. In fact, the registration database and the persistence interface are located on the client computer running the editor. This becomes clear by examining Figure 1 and the Abstract of Massena, which indicate that the design time control, and consequently the persistence interface and the registration database used to identify a DTC, are running at design-time as part of the editor on the client, not at run time. The cited reference therefore discloses storing data on the client, whereas applicant claims storing data on the server.

The examiner also states that Massena discloses assigning unique identifiers for data objects. However, as previously discussed, applicant's reading of Massena column 7, line 60 to column 8, line 10, reveals no support for the examiner's position. Instead, the cited portion appears to disclose information on identifying DTCs based on an ID. In contrast, applicant claims "assigning a unique identifier" which is quite different than identifying a DTC based on an ID

The examiner asserts that Massena teaches that a document is generated dynamically with embedded unique identifiers. However, applicant submits that the examiner is in error reaching that conclusion. In fact, generating a document in Massena happens statically at design time. This is clear from a review of column 3, lines 6-7 of Massena, which states that processing is performed when files are saved rather than when the file is accessed by the web browser. The cited reference therefore discloses static embedding. In contrast, applicant claims that embedding happens dynamically.

The examiner further states that Massena discloses analyzing requests for unique identifiers. However, applicant submits that the examiner is in error reaching that conclusion. Massena does not teach or suggest analyzing a page request, but instead, identifying a DTC. The cited reference therefore does not appear to disclose analyzing requests for unique identifiers.

In addition, the examiner asserts that Massena discloses a method of analyzing a request sent by the browser for unique identifiers, citing column 7, lines 60-67 and column 8, lines 1-10 in support of his assertion. However, applicants' reading of the cited portion of Massena reveals no support for the examiner's position. Instead, the cited portion appears to describe identifying DTCs during editing on the client computer, which appears irrelevant to analyzing a request sent by the web browser.

Claim 17

Claim 17 is dependent from claim 16, and for all the reasons stated with reference to claim 16, applicant submits that claim 17 is patentably distinct from the cited reference.

The examiner asserts that Massena teaches, in column 7, lines 36-45, that assigned unique identifiers are embedded in a uniform resource locator. However applicant's reading of the cited portion of Massena reveals no support for the examiner's position. Instead, Massena reveals that various items can be generated by a DTC and that a DTC can be downloaded as needed. Applicant might conclude that downloading involves using a uniform resource locator, but there is no teaching or suggestion in Massena that the URL should contain a newly assigned unique identifier. The cited position therefore appears irrelevant to the claim.

Claim 18

Claim 18 is dependent from claim 16, and for all the reasons stated with reference to claim 16, applicant submits that claim 18 is likewise patentably distinct from the cited reference.

The examiner asserts that Massena discloses unique identifiers embedded in scripts contained in the document, citing column 7, lines 36-45 in support of his assertion. However applicant's reading of Massena reveals no support for the examiner's position. Instead, as with claim 17, Massena at most discloses that various items can be generated by a DTC and the DTC downloaded as needed. In particular, the cited portion states that scripts can be contained in the

generated document, but does not disclose that the unique identifiers generated are contained in scripts.

Claim 19

Claim 19 is dependent from claim 16, and for all the reasons stated with reference to claim 16, applicant submits that claim 19 is likewise patentably distinct from the cited reference.

The examiner asserts that Massena discloses the use of identifiers which are unique within a session, citing column 7, line 60 to column 8, line 10. However, applicant submits that the examiner is in error reaching that conclusion. In fact, the ID mentioned by Massena is static, not dynamic. It is not unusual for a single page template to occur more than once within a session, and therefore all static identifiers contained on the page template would also recur. Consequently, the identifiers are not unique within a session since they recur. In contrast, applicant's claim recites that identifiers are unique within a session.

Claim 20

Claim 20 is dependent from claim 16, and for all the reasons stated with reference to claim 16, applicant submits that claim 20 is patentably distinct from the cited reference.

The Examiner did not cite any portion of Massena in support of his assertion that the claim is anticipated, but for similar reasons as discussed with reference to claims 17-19, applicant submits that Massena does not teach or suggest the use of an identifier which is unique within all pages generated during a specified time.

Claim 21

Claim 21 is dependent from claim 16, and for all the reasons stated with reference to claim 16, applicant submits that claim 21 is patentably distinct from the cited reference.

Claim 22

Claim 22 is an independent claim directed to a computer having an editor and a page generator. The examiner asserts that Massena (Fig. 1 and Abstract) teaches an editor which is operable within the web browser. However, applicant respectfully disagrees. As previously



noted, the editor of Massena runs as a separate program. This is clearly depicted in Fig. 1, which illustrates the editor and the browser box. In contrast, applicant's claim recites that the editor is operable within the web browser.

The examiner asserts that Massena (col. 3, lines 30-35; col. 11, lines 60-65) discloses a method for generating a document from a document template. However, applicant respectfully disagrees. In Massena, HTML generation takes place by adding the HTML code to the document being edited, thus transforming a document template. In contrast, applicant's claim recites generating an entirely new document, from a document template.

Claim 22 was amended to explicitly include the case wherein the browser runs on the same computer as the page generator. This is useful as indicated on page 26 line 18 in the application. Applicant submits that the claim is patentably distinct over the cited reference for all the reasons stated above.

Claim 23

Claim 23 is dependent from claim 22, and for all the same reasons, applicant submits that claim 23 is patentably distinct from the cited reference. Applicant has amended claim 23 to be consistent with the amendment of claim 22.

In addition, the examiner asserts that Massena (column 7, lines 30-35) shows an editor that operates functional applications in an edit mode permitting editing directly in the web browser. Applicant's reading of the cited portion reveals no support for the examiner's assertion. The cited portion appears to discuss the advantages of generating HTML text to work independent of the type of browser being run on the client computer. This is quite different from editing inside the web browser such that the editor is operating a functional application in an editing mode. The cited portion therefore appears irrelevant to the claim.

Applicant submits that this claim is patentably distinct over the cited reference for the reasons stated above.

Claim 24

Claim 24 is dependent from claim 23, which is in turn dependent from claim 22, and for all the same reasons as discussed for those claims, applicant submits that claim 24 is patentably

distinct from the cited reference. Applicant amended the claim to be consistent with the generalization of claim 22 adding a new restriction “reacting on a browser request”, because the restriction stating “on the server” is not meaningful in the special case of the browser running on a single computer.

In addition, the examiner asserts that Massena (Abstract; Fig. 1; column 5, lines 7-10) shows components that can react on user responses by executing instructions on the server. However, applicant respectfully disagrees. The cited portion appears to disclose the process of saving and transforming a page, and possibly inserting server side scripting, but does not, however, elaborate on the scripts generated. Server side scripts are normally executed on page generation taking place **before** display of a component. In contrast, applicant’s claim (prior to amendment) recited components that react, upon user interaction (which necessarily happens **after** display of the component), by executing instructions on the server.

Applicant submits that the generalized claim is patentably distinct over the cited reference because the arguments stated above stay valid.

Claim 25

Claim 25 is dependent from claim 24, which is in turn dependent from claims 23 and 22, and for all the same reasons as discussed for those claims, applicant submits that claim 25 is likewise patentably distinct from the cited reference. Applicant amended the claim to be consistent with the generalization of claim 22 further restricting page generation to take place upon a request from the web browser.

The examiner asserts that Massena shows a data store of component classes on the server. As discussed for claim 6 above, applicant respectfully disagrees and submits that Massena does not teach or suggest this element.

The examiner also says that Massena (column 11, lines 50-52) discloses a server computer containing a parser able to detect components marked on page templates. Applicant respectfully disagrees and submits that Massena parses OBJECT tags as part of the editor on the client computer. *See* column 11, line 53, and Fig. 1. Fig. 1 of Massena clearly shows that the editor is located on the client computer and not on the server computer.

The examiner also asserts that Massena (column 5, lines 50-55) discloses an editor capable of showing a menu of components available for insertion into the page template. However, the cited portion relates to a property browser and property page frame, not a menu of components as recited in applicant's claim.

Claim 25 is dependent from claim 24, which is in turn dependent from claims 23 and 22, and for all the same reasons as discussed for those claims, applicant submits that also the claim 25 that depends on the generalized claim 22 is likewise patentably distinct from the cited reference. In addition the argument about the menu of components stays valid with the generalized claim 22. The additional restriction in the claim further assures that page generation using components takes place upon request of the browser and not inside the editor as it appears to be the case in Massena.

Claim 26

Claim 26 is an independent claim directed to a system for modifying documents on a server. The examiner asserts that Massena shows a data store on the server, citing column 8, line 7 and column 9, lines 55-60 in support of his assertion. However, applicant submits that the examiner is in error reaching that conclusion. As previously discussed, the registration database and the persistence interface are located on the client computer running the editor, not on the server computer. (See Massena Fig. 1, Abstract). Further, the design time control and consequently the persistence interface and the registration database used to identify a DTC are running at design-time, not at run-time, as part of the editor on the client computer.

The examiner has also asserted that Massena discloses a program that transforms a first document into a second document having features which permit editing of the first document such that at least part of the second document appears and functions similar to the first document, particularly the use of OLE. However, applicant's review of the abstract does not reveal any support for the examiner's position. Instead, the cited portion of Massena appears to disclose that DTCs can make use of various features using OLE. The abstract does not teach or suggest the transformation of one document into another document, but to the contrary, Massena's editor performs transformations on a single document. In contrast, applicant's claim 26 expressly

recites transforming the first document into a second document that, when displayed, permits editing of the first document.

The examiner also asserts that Massena discloses a second program on the server which includes instructions to modify documents on the server. However, applicant's reading of the cited portions of Massena reveals no support for the examiner's position. Instead, the cited portions appear to describe various programs running on the client computer. In contrast, applicant explicitly recites that the first and second programs run on the server.

Claim 27

Claim 27 is dependent from claim 26. For all the reasons already discussed regarding claim 26 above, applicant submits that claim 27 is likewise not taught or suggested by Massena.

In addition, the examiner does not specify an element in Massena that corresponds to the "second document" as claimed by applicant. Applicant submits that there is no teaching or suggestion in Massena of a second document, and therefore, the claim is believed to be patentable over the cited reference.

Claim 28

Claim 28 is dependent from claim 27, which is in turn dependent from claim 26. For all the reasons already discussed regarding claims 26-27 above, applicant submits that claim 28 is likewise not taught or suggested by Massena.

In addition, the examiner does not specify an element in Massena that corresponds to the "second document" as claimed by applicant. Applicant submits that there is no teaching or suggestion in Massena of a second document, and therefore, the claim is believed to be patentable over the cited reference.

Claim 29

Claim 29 is dependent from claim 28, which is in turn dependent from claim 27 and claim 26. For all the reasons already discussed regarding claims 26-28 above, applicant submits that claim 29 is likewise not taught or suggested by Massena.

Claim 30

Claim 30 is dependent from claim 26. For all the reasons already discussed regarding claim 26 above, applicant submits that claim 30 is not taught or suggested by Massena.

In addition, the examiner does not specify an element in Massena that corresponds to the “feature” as claimed by applicant. Applicant submits that there is no teaching or suggestion in Massena of a feature as claimed by applicant, and therefore, the claim is believed to be patentable over the cited reference.

Claim 31

Applicant has amended claim 31 to be dependent from claim 30, and for all the reasons discussed regarding claim 30 above, applicant submits that claim 31 is not taught or suggested by Massena.

Claim 32

Claim 32 is dependent from claim 26, and for all the reasons discussed regarding claim 26 above, applicant submits that claim 32 is not taught or suggested by Massena.

Claim 33

Applicant has amended claim 33 to reverse the recitation of the first document and the second document, to correct the erroneous reference. Claim 33 is dependent from claim 32, and for all the reasons discussed regarding claim 32 above, applicant submits that claim 33 is likewise not taught or suggested by Massena.

Claim 34

Claim 34 is an independent claim directed to a method for generating a page from a page template having components. Applicant has amended claim 34, not to avoid the reference, but to correct the claim and make explicit the fact that pages are generated for display by a browser. Applicant also clarified that the first step is done for all components denoted on the page template and that the second step happens based on a request initiated by the browser.

Furthermore, it was clarified that the object created is representing the component. Finally the “or” was replaced by other wording and additional dependent claims.

The examiner asserts that Massena (col. 8, lines 1-10; col. 5, lines 25-30) discloses a method for generating a page from a page template containing components. However, applicant respectfully disagrees. In Massena, HTML generation takes place by adding the HTML code to the document being edited, thus transforming a document template. In contrast, applicant’s claim recites generating an entirely new document, i.e. a newly generated page, from a page template.

The amended claim states that storing objects representing the components takes place based on the data contained in a request initiated by the browser, which means especially after the browser sent the request, at run-time. In contrast DTCs in Massena work during editing. The discussion regarding claim 10 contains detail about storing data and representing the components.

Claim 35

Claim 35 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 35 is not taught or suggested by Massena. In addition, the examiner asserts that Massena (column 5, lines 58-62) discloses that the constructor method is used to generate browser code and that it is called during page generation. However, applicants’ reading of the cited portion reveals no support for the examiner’s position. Instead, the cited portion appears to disclose how controls are created. In contrast, applicant claims that browser code is generated by the constructor and that objects are actually created during page generation.

Claim 35 was amended to be dependent on Claim 44 that was newly added when amending claim 34.

Claim 36

Claim 36 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 36 is likewise not taught or suggested by Massena. In addition, the examiner asserts that Massena (col. 7, lines 60-67; col. 8, lines 1-10) discloses that component

objects are looked up in session memory. However, applicant submits that the cited portion discloses how DTCs are looked up during editing on the client.

From the use of the term “session memory” in the present patent application, it is clear that this phrase refers to data stored on the server. Therefore, applicant’s claim inherently references a look up process during page generation upon request of a web browser, which is quite different to a look up process on the client during editing.

Also, the examiner interprets the name attribute as ID of a DTC. However, applicant submits that the ID corresponds to the name of a component class but not to the name attribute of a particular component instance.

Claim 37

Claim 37 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 37 is likewise not taught or suggested by Massena. Applicant has amended claim 37 as shown in Appendix A, not for the purpose of distinguishing the reference, but to generalize the nature of the components and to avoid any ambiguity associated with the term “interactive.”

The examiner asserts that Massena (col. 7, lines 60-67; col. 8, lines 1-10) discloses a method for generating a unique identifier for all interactive components. However, applicant respectfully disagrees. Instead, the cited portion appears to disclose identifying DTCs using the ID. In contrast, applicant’s claim says that the unique identifiers are generated during page generation.

Claim 38

Claim 38 is dependent from claim 37, and for all the reasons discussed therein, applicant submits that claim 38 is likewise not taught or suggested by Massena. Applicant has amended claim 38 as shown in Appendix A, not for the purpose of distinguishing the reference, but as in claim 37, to generalize the nature of the components and to avoid any ambiguity associated with the term “interactive.”

The examiner asserts that Massena (col. 5, lines 25-35) discloses inserting certain objects into a list. However, the cited portion appears to discuss the use of DTCs as objects, but does not teach inserting these objects into any list.

The examiner also cites Massena (col. 3, lines 45-50) as disclosing calling a method for one of the objects whose unique identifier is contained in the form data set and in the list of listening components. However, the cited portion appears related to calling a method, but does not teach or suggest anything about the objects whose method is called.

Also, applicant has amended the claim to replace the term “bid” with the term “unique identifier” to be consistent with the other claims.

Claim 39

Claim 39 is dependent from claim 34, and for all the reasons discussed for claim 34, applicant submits that claim 39 is not taught or suggested by Massena.

The examiner asserts that Massena (col. 7, lines 60-67; col. 8, lines 1-10) discloses evaluating the attributes of a component. However, the cited portion appears to disclose information on identifying a DTC, which is irrelevant to the evaluation of attributes as recited in the present claims.

The examiner also says that Massena (col. 9, lines 30-65; col. 10, lines 15-30; col. 7, lines 60-67; col. 8, lines 1-10) teaches calling the constructor method of the component class, wherein the constructor method generates browser code. However, applicant submits that Massena discloses that a non-constructor method is called to generate run-time text. (*See* col. 10, lines 15-30)

The examiner cites Massena at col. 5, lines 58-62, as disclosing object initialization inside the algorithm. However, the cited portion appears to disclose some information on initialization but no information on when it happens.

The examiner also cites Massena at col. 5, lines 25-30, as disclosing “repeating these steps for each node.” However, applicant submits that the cited portion discloses that web page content can be repeated on several pages by using a single component several times. In contrast, applicant’s claim inherently includes a loop in a certain algorithm.

Claim 40

Claim 40 is dependent from claim 39, and for all the reasons discussed therein, applicant submits that claim 40 is likewise not taught or suggested by Massena. In addition, as previously noted with regard to claim 4 above, Massena fails to teach or suggest nested components.

The examiner also cites Massena at col. 5, lines 25-30, as disclosing recursively performing the method for all nodes nested inside the component. However, applicant respectfully disagrees. The cited portion of Massena appears to disclose that web page content can be repeated on several pages by using a single component several times without the use of an algorithm. In contrast, applicant expressly recites a recursive method.

V. NEW CLAIMS 41-124 ARE PATENTABLE OVER THE CITED REFERENCE

Applicant has added new claims 41-124 and submits that the newly added claims are patentably distinct over the cited reference.

Claim 41 is a new claim dependent on claim 1 and, for the same reasons, is patentably distinct over the cited reference. This claim emphasizes that the editor has a client part.

Claim 42 is a new claim dependent on claim 41, explicitly stating that the client part comprises instructions that are automatically downloaded before editing. In contrast, Massena's invention seems to require an editor installed on the client.

Claim 43 is a new claim dependent on claim 26 which recites automatic downloading of script to the client. In contrast, Massena's invention seems to require an editor installed on the client.

Claim 44 is a new claim dependent on claim 34 and, for the same reasons, is patentably distinct over the cited reference. The claim recites creating new data objects based on a request representing a component. Massena's DTCs in contrast work during editing as part of the editor.

Claim 45 is a new claim dependent on claim 34 and, for the same reasons, is patentably distinct over the cited reference. It introduces tag syntax.

Claim 46 is a new claim dependent on claim 45 explicitly stating that the tag name identifies a component class. In contrast, Massena seems to use OBJECT-tags to denote a component, probably using tag-attributes to identify the component class.

Claim 47 is a new claim dependent on claim 36 and, for the same reasons, is patentably distinct over the cited reference. It introduces tag syntax, as in claim 45.

Claim 48 is a new claim dependent on claim 36 and, for the same reasons, is patentably distinct over the cited reference. The cited reference does not appear to disclose reusing an object based on the data in a page request.

Claim 49 is a new claim dependent on claim 36 and, for the same reasons, is patentably distinct over the cited reference. The cited reference does not appear to disclose storing an object under a name in session memory, as discussed with reference to claim 36.

Claim 50 is a new claim dependent on claim 49 and, for the same reasons, is patentably distinct over the cited reference. It introduces creation of new objects for components denoted on a page without having a name attribute. Please refer to the discussion of claims 34 and 44 for creating new objects based on a request. In addition the cited reference does not seem to disclose components denoted on a page with or without a name attribute.

Claim 51 is a new independent claim. It introduces a system for editing components on web document templates that runs itself as a web application.. Instead Massena seems to require a client based editor and does not teach second documents.

Claim 52 is a new claim dependent on claim 51 and is therefore patentably distinct over the cited reference.

Claim 53 is a new claim dependent on claim 52 and, for the same reasons, is patentably distinct over the cited reference. In addition, the claim requires the first program to run on a client and the second program to run on a server computer. In contrast, the editing functionality of Massena is running on the client computer.

Claim 54 is a new claim dependent on claim 52 and for the same reasons, is patentably distinct over the cited reference.

Claim 55 is a new claim dependent on claim 52 and is therefore patentably distinct over the cited reference.

Claim 56 is a new claim dependent on claim 52 and is therefore patentably distinct over the cited reference. In addition, the claim requires document generation that uses components to take place dynamically on request of the first software program. In contrast, Massena's components generate code statically during editing.

Claim 57 is a new claim dependent on claim 56 and is therefore patentably distinct over the cited reference. In addition, it introduces edit features as discussed after Claim 1.

Claim 58 is a new claim dependent on claim 56 and is therefore patentably distinct over the cited reference.

Claim 59 is a new independent claim. The claim emphasizes the central idea of the invention, i.e., allowing the user to work on the generated document instead of the document template.. In contrast, Massena's figure 1 shows two views of the document: the document template for editing in the editor and the generated document for viewing in the browser. This is clearly different from a single view that can be edited.

Claim 60 is a new claim dependent on claim 59 and is therefore patentably distinct over the cited reference. In addition the claim introduces a data network, and requires the document generator to run on the server. In contrast in Massena the document generator runs on the client.

Claim 61 is a new claim dependent on claim 60 and is therefore patentably distinct over the cited reference.

Claim 62 is a new claim dependent on claim 60 and is therefore patentably distinct over the cited reference. In addition the claim requires a web browser to display said view. Massena seems to use two views, one in the editor and another one in the browser. Applicant believes that editing takes place in Massena's invention by clicking on the editor. This is different from clicking on the browser.

Claim 63 is a new claim dependent on claim 60 and is therefore patentably distinct over the cited reference. The cited reference does not seem to mention that Massena's invention performs automatic reloads.

Claim 64 is a new claim dependent on claim 59 and is therefore patentable distinct over the cited reference.

Claim 65 is a new claim dependent on claim 64 and is therefore patentably distinct over the cited reference. It is similar to claim 62.

Claim 66 is a new claim dependent on claim 64 and is therefore patentably distinct over the cited reference.

Claim 67 is a new claim dependent on claim 59 and is therefore patentably distinct over the cited reference.

Claim 68 is a new claim dependent on claim 59 and is therefore patentably distinct over the cited reference. Edit information is discussed with claim 1.

Claim 69 is a new claim dependent on claim 68 and is therefore patentably distinct over the cited reference. In addition the claim introduces the use of edit information.

Claim 70 is a new claim dependent on claim 69 and is therefore patentably distinct over the cited reference. In addition the claim introduces position information on components. The cited reference does not seem to mention position information.

Claim 71 is a new claim dependent on claim 59 and is therefore patentably distinct over the cited reference. It is similar to claim 62.

Claim 72 is a new claim dependent on claim 71 and is therefore patentably distinct over the cited reference. It introduces performing a reload in the browser. The cited portions of Massena do not seem to disclose issuing a reload in the browser.

Claim 73 is a new claim dependent claim 59 and is therefore patentably distinct over the cited reference.

Claim 74 is a new independent claim. The claim is about the central idea of the invention to, upon a document request, dynamically decide on the components included in the generated document. "Dynamic" means that the components displayed on the generated document can differ with every document request in number, size or other characteristic, from the components initially included on the document template. In contrast Massena seems to generate documents statically during or after editing.

Claim 75 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. It clarifies edit function.

Claim 76 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. It introduces tag syntax as claim 46.

Claim 77 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. It introduces a data network.

Claim 78 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. It introduces excluding a component from the generated document.



Claim 79 is a new claim dependent on claim 78 and is therefore patentably distinct over the cited reference

Claim 80 is a new claim dependent on claim 79 and is therefore patentably distinct over the cited reference

Claim 81 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference.

Claim 82 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference

Claim 83 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference.

Claim 84 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference.

Claim 85 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference.

Claim 86 is a new claim dependent on claim 85 and is therefore patentably distinct over the cited reference.

Claim 87 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. It introduces excluding nested components from the generated document.

Claim 88 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference

Claim 89 is a new claim dependent on claim 74 and is therefore patentably distinct over the cited reference. The claim is about the advantage of ISSC that a component needs just one routine to display itself. To applicant it seems that Massena's DTCs need one routine to display the component during editing and another routine the generate browser code to display the component at run-time.

Claim 90 is a new independent claim focusing on the ability of the invention to edit functional documents by using the web browser as editor. Editing functional documents is important for the invention to allow simultaneous debugging and editing. In contrast as shown by Massena's figure 1 the editor is not running inside the browser in the cited reference.

Claim 91 is a new claim dependent on claim 90 and is therefore patentably distinct over the cited reference. Massena seems not to use a second browser window for displaying information.

Claim 92 is a new claim dependent on claim 90 and is therefore patentably distinct over the cited reference.

Claim 93 is a new claim dependent on claim 92 and is therefore patentably distinct over the cited reference. In addition edit features for interpretation of the browser are introduced. These are not present in the cited reference as discussed with reference to claim 1.

Claim 94 is a new claim dependent on claim 93 and is therefore patentably distinct over the cited reference.

Claim 95 is a new claim dependent on claim 94 and is therefore patentably distinct over the cited reference. In addition the claim requires the document modification and the document generation to happen on the server computer. In contrast these are done on the client computer in Massena's invention.

Claim 96 is a new claim dependent on claim 90 and is therefore patentably distinct over the cited reference. The cited reference does not give information on links during editing.

Claim 97 is a new independent claim. It introduces components for execution on the server. In addition it is explicitly states that document generation takes place based on the data contained in a document request. In contrast Massena's DTCs generate code during editing.

Claim 98 is a new claim dependent on claim 97 and is therefore patentably distinct over the cited reference. It introduces analyzing a request. Massena seems not to teach this explained in detail after Claim 11.

Claim 99 is a new claim dependent on claim 98 and is therefore patentably distinct over the cited reference.

Claim 100 is a new claim dependent on claim 99 and is therefore patentably distinct over the cited reference. It introduces tag syntax as claim 46.

Claim 101 is a new claim dependent on claim 99 and is therefore patentably distinct over the cited reference. It introduces a component containing another component. According to applicants reading of the cited reference Massena does not teach nested components as explained in detail after Claim 4.



Claim 102 is a new claim dependent on claim 101 and is therefore patentably distinct over the cited reference. It introduces deciding how often to insert the second component into the generated document. Massena seems to teach a static component set instead as explained in detail after Claim 5.

Claim 103 is a new claim dependent on claim 98 and is therefore patentably distinct over the cited reference. It introduces deciding how often to insert a component into the generated document. Massena seems to teach a static component set instead as explained in detail after Claim 5.

Claim 104 is a new claim dependent on claim 103 and is therefore patentably distinct over the cited reference.

Claim 105 is a new claim dependent on claim 98 and is therefore patentably distinct over the cited reference.

Claim 106 is a new claim dependent on claim 105 and is therefore patentably distinct over the cited reference.

Claim 107 is a new claim dependent on claim 98 and is therefore patentably distinct over the cited reference.

Claim 108 is a new claim dependent on claim 107 and is therefore patentably distinct over the cited reference.

Claim 109 is a new claim dependent on claim 98 and is therefore patentably distinct over the cited reference

Claim 110 is a new claim dependent on claim 109 and is therefore patentably distinct over the cited reference.

Claim 111 is a new claim dependent on claim 110 and is therefore patentably distinct over the cited reference.

Claim 112 is a new claim dependent on claim 109 and is therefore patentably distinct over the cited reference. It introduces dynamically assigning unique identifiers. Massena seems not to teach this as explained in detail after Claim 16. In addition claim 97 states that assigning happens based on the data contained in a document request, while Massena's invention generates code statically during editing.

Claim 113 is a new claim dependent on claim 112 and is therefore patentably distinct over the cited reference. It introduces analyzing a request. Massena seems not to teach this explained in detail after Claim 11.

Claim 114 is a new independent claim. The claim is about components executing on the server and still cooperating with the editor. In contrast Massena's DTCs run on the client.

Claim 115 is a new claim dependent on claim 114 and is therefore patentably distinct over the cited reference.

Claim 116 is a new claim dependent on claim 115 and is therefore patentably distinct over the cited reference.

Claim 117 is a new claim dependent on claim 115 and is therefore patentably distinct over the cited reference.

Claim 118 is a new claim dependent on claim 115 and is therefore patentably distinct over the cited reference.

Claim 119 is a new claim dependent on claim 114 and is therefore patentably distinct over the cited reference.

Claim 120 is a new claim dependent on claim 119 and is therefore patentably distinct over the cited reference.

Claim 121 is a new claim dependent on claim 114 and is therefore patentably distinct over the cited reference.

Claim 122 is a new claim dependent on claim 121 and is therefore patentably distinct over the cited reference. It introduces a page for editing components attributes. The cited reference does not teach a page for editing attributes since Massena's DTCs run on a client,

Claim 123 is a new claim dependent on claim 114 and is therefore patentably distinct over the cited reference. It introduces tag syntax as claim 46.

Claim 124 is a new claim dependent on claim 114 and is therefore patentably distinct over the cited reference. It introduces the same feature as claim 89.

Claim 125 is a new independent claim. It introduces a method to edit applications emphasizing the idea of working on generated pages. The cited reference does not appear to teach or suggest the selecting and identifying steps as claimed.

Claim 126 is a new claim dependent on claim 125 and is therefore patentably distinct for the same reasons.

Claim 127 is a new claim dependent on claim 125 and is therefore patentably distinct for the same reasons.

VI. DRAWING CORRECTIONS

Applicant requests approval of the drawing corrections submitted as Appendix C herewith.

VII. PRIORITY DATA

Applicant has concurrently filed a request to correct the official filing receipt to reflect priority from applicant's prior U.S. provisional applications. A copy of the request is included as Appendix D.


VIII. CONCLUSION

For all the foregoing reasons, applicant submits that the claims are in condition for allowance, and the examiner's favorable reconsideration to that end is solicited. If additional questions remain, the examiner is encouraged to telephone the undersigned.

Respectfully submitted,

DERGOSITS & NOAH LLP

Dated: 3/25, 2002

By: 
Richard A. Nebb
Reg. No. 33,540

Four Embarcadero Center, Suite 1150
San Francisco, California 94111
(415) 705-6377 tel
(415) 705-6383 fax
mnebb@dergnoah.com